

Building non-coding RNA families

Lars Barquist¹, Sarah W. Burge¹, and Paul P. Gardner²

¹Wellcome Trust Sanger Institute, Hinxton, UK

²University of Canterbury, Christchurch, NZ

March 8, 2013

Abstract

Homology detection is critical to genomics. Identifying homologous sequence allows us to transfer information gathered in one organism to another quickly and with a high degree of confidence. Non-coding RNA (ncRNA) presents a challenge for homology detection, as the primary sequence is often poorly conserved and de novo structure prediction remains difficult. This chapter introduces methods developed by the Rfam database for identifying “families” of homologous ncRNAs from single “seed” sequences using manually curated alignments to build powerful statistical models known as covariance models (CMs). We provide a brief overview of the state of alignment and secondary structure prediction algorithms. This is followed by a step-by-step iterative protocol for identifying homologs, then constructing an alignment and corresponding CM. We also work through an example, building an alignment and CM for the bacterial small RNA MicA, discovering a previously unreported family of divergent MicA homologs in *Xenorhabdus* in the process. This chapter will provide readers with the background necessary to begin defining their own ncRNA families suitable for use in comparative, functional, and evolutionary studies of structured RNA elements.

1 Introduction

Alignment is a central problem in bioinformatics. A wide range of critical applications in genomics rely on our ability to produce “good” alignments. Single-sequence homology search as implemented in tools such as BLAST[1] is an (often heuristic) application of alignment. The sensitivity and specificity of homology search can be improved by the use of evolutionary information in the form of accurate substitution and insertion-deletion (indel) rates derived from multiple sequence alignments (MSAs), captured in the statistical models used by HMMER[2] and Infernal[3] for protein and RNA alignments respectively. These models can be interpreted as defining “families” of homologous sequences, as in the Pfam and Rfam databases[4, 5]. By using these models to classify sequences, we can infer functional and structural properties of uncharacterized sequences.

Unfortunately, producing the high-quality “seed” alignments of RNA these methods require remains difficult. While proteins can be aligned accurately using only primary sequence information with pairwise sequence identities as

low as 20% for an average-length sequence[6, 7], it appears that the “twilight zone” where blatantly erroneous alignments occur between RNA sequences may begin at above 60% identity[8]. The inclusion of secondary structure information can improve alignment accuracy[9], but predicting secondary structure is not trivial[10]. An instructive example of the difficulties this can lead to is the case of the 6S gene, a bacterial RNA which modulates σ^{70} activity during the shift from exponential to stationary growth. The *Escherichia coli* 6S sequence was determined in 1971[11] and its function determined in 2000[12]. However, the extent of this gene’s phylogenetic distribution was not realized until 2005 when Barrick and colleagues carefully constructed an alignment from a number of deeply diverged putative 6S sequences, and through successive secondary-structure aware homology searches demonstrated its presence across large swaths of the bacterial phylogeny[13]. Even now, new homologs are discovered on a regular basis[14, 15], and 6S appears to be an ancient and important component of the bacterial regulatory machinery.

It is our hope to make these techniques accessible to sequence analysis novices. This chapter aims to introduce the techniques necessary to construct a high-quality RNA alignment from a single seed sequence, and then use the information contained in this alignment to identify additional more distant homologs, expanding the alignment in an iterative fashion. These methods, while time-consuming, can be far more sensitive than a BLAST search[16]. We will briefly review the state of the art in RNA sequence alignment and structure prediction. We then present a brief protocol which starts with a single sequence, and then uses a collection of web and command-line based tools for alignment, structure prediction, and search to construct an Infernal covariance model (CM), a probabilistic model which captures many important features of structured RNA sequence variation[3]. These models may then be used in the iterative expansion of alignments or for homology search and genome annotation. CMs are also used by the Rfam database in defining RNA sequence families, and are the subject of a dedicated RNA families track at the journal *RNA Biology*[17]. Finally, we present as an instructive example the construction of an RNA family for the enterobacterial small RNA MicA, discovering a convincing divergent clade of homologs in the process.

1.1 RNA Alignment and Secondary Structure Prediction

RNA sequence alignment remains a challenge despite at least 25 years of work on the problem. As discussed above, alignments based on primary sequence become highly untrustworthy below 60% pair-wise sequence identity, likely due to the lower information content of individual nucleic acids as compared to amino acids in protein alignments. This can be intuitively understood by recalling the fact that 3 nucleic acids are required to encode an individual amino acid; so, an amino acid carries 3 times as much information as a nucleic acid (a bit less, actually, due to the redundancy of the genetic code). In addition, the larger alphabet size of protein sequences allows for the easy deployment of more complex substitution models, and a glut of protein sequence data allows for highly effective parameterization of these models.

The incorporation of secondary structure, i.e. base-pairing, information has been proposed as a means to make up for these difficulties in RNA alignment methods. The first proposal for such a method is now known as the Sankoff

algorithm[18]. The Sankoff algorithm uses dynamic programming, an optimization technique long central to sequence analysis¹. Dynamic programming had previously been applied to the problems of sequence alignment[22] and RNA folding[23]. Sankoff proposed a union of these two methods. Unfortunately, the resulting algorithm has a time requirements of $\mathcal{O}(L^{3N})$ and space requirements of $\mathcal{O}(L^{2N})$ where L is the sequence length and N is the number of sequences aligned. This is impractical, even for small numbers of short sequences. A number of faster algorithms have been developed to approximate Sankoff alignment. Recent examples include CentroidAlign[24], mLocARNA[25], and FoldalignM[26]. These methods can push the RNA alignment twilight zone as low as 40 percent identity[8].

However, for the purpose of family-building, we are often starting with a single sequence of unknown secondary structure, and have to gather additional homologs using a fast alignment tool, such as BLAST. Such methods are not able to reliably detect homologs below 60 percent sequence identity. In this range of pair-wise sequence identities, the slight increases in accuracy of Sankoff-type algorithms over non-structural alignment is only rarely worth the additional computational costs involved². Alignments generated with standard alignment tools can then be used as a basis for predictions of secondary structure using tools like Pfold[28], RNAalifold[29], or CentroidFold[30].

Regardless, all modern alignment tools, Sankoff-type or standard, suffer from a number of known problems. Most alignment tools use *progressive alignment*. This means that the aligner decomposes the alignment problem in to a series of pair-wise alignment problems along a guide tree. This greatly reduces the computational complexity of the alignment problem, but means that any error in an early pair-wise alignment step is propagated through the entire alignment. A number of solutions have been proposed to this problem, such as explicitly modeling insertion-deletion histories[31] or using modified or alternative optimization methods such as consistency-guided progressive alignment[32] or sequence annealing[33]. A second issue is that it is not clear which function of the alignment aligners should be optimizing, and many appear to over-predict homology[34, 27, 35]. Finally, many parameters commonly used in alignment, such as gap opening and closing probabilities and substitution matrices, appear to vary across organisms, sequences, and even positions within an alignment. All of this leads to considerable uncertainty in alignment[36], which is not easily captured by most current alignment methods. The additional parameters introduced by RNA secondary structure prediction only compounds these these problems.

A final problem with alignment is the issue of determining whether two sequences are similar due to *homology* or *analogy*. Homology describes a similarity in features based on common descent; for instance, all bird wings are homologous wings. Analogy, on the other hand, describes a similarity in features based on common function without common descent; bat and bird wings perform the

¹A full explanation of dynamic programming is beyond the scope of this book chapter, but for a brief introduction see two excellent primers by Sean Eddy covering applications to alignment[19] and secondary structure prediction[20]; for those seeking a deeper understanding Durbin *et al.*[21] provides coverage of dynamic programming as well as covariance models.

²For recent benchmarks of alignment tools on ncRNA sequences see Hamada *et al.*[24] and the supplementary information of Bradley *et al.*[27]; Hamada includes comparisons of aligner runtimes, while Bradley examines relative performance over a range of pair-wise sequence percent identities.

same function, and appear superficially similar. However, their evolutionary histories are quite different. In sequence analysis, we often assume that aligned residues within an alignment share common ancestors, but this assumption can be confounded by analogous sequence. These analogs often take the form of *motifs*, short sequences which perform specific functions within the RNA molecule and can arise easily through convergent evolution. An example of such a motif is the bacterial rho-independent terminator[37], a short hairpin responsible for halting transcription in many species. While such motifs can be a boon in discovering novel ncRNA genes[38] or aligning homologs which contain them, they can also be a source of false-positives when attempting to build an alignment of homologous sequences.

Rfam has developed a pipeline designed to address many of these problems[39]. Starting from a single sequence, we iteratively expand an alignment using Infernal covariance models. During each iteration, we use a variety of automatic alignment and secondary structure prediction tools together with manual curation and editing in an effort to avoid many of the issues raised above. While the Rfam pipeline is designed to run on a high-end computational cluster, we have adapted the process here to make it accessible to anyone with a commodity PC and an internet connection.

2 Materials

2.1 Single Sequence Search

We rely on NCBI BLAST[1] to quickly identify close homologs of RNA sequences in this protocol. NCBI and EMBL-EBI both maintain servers[40, 41] with slightly different interfaces, though there are no substantive differences in the implementations. We use the NCBI server here. EBI also maintains servers for a number of BLAST and FASTA derivatives, which may be helpful. Both sites also allow users to BLAST against databases of expressed sequences including GEO at NCBI, and high throughput cDNA and transcriptome shotgun assembly databases at EMBL-EBI. Such searches can be helpful for gathering comparative expression data for your ncRNA.

A nucleotide version of the HMMER3 package[2] for sequence search is currently in development which promises both increased sensitivity and specificity over BLAST at little additional computational cost. We hope that a web server similar to the one currently available for protein sequences[42] will be forthcoming. If it is possible that homologous sequences are spliced (e.g. introns in the U3 snoRNA[43]), then a splice-site aware search method may be useful, such as BLAT[44] or GenomeWise[45], but there are not publicly available web servers for them that we are aware of.

Resource	Reference	URL
NCBI-BLAST	[40]	http://blast.ncbi.nlm.nih.gov/Blast.cgi
EMBL-EBI NCBI-BLAST	[41]	http://www.ebi.ac.uk/Tools/sss/ncbiblast/
EMBL-EBI Sequence Search	[41]	http://www.ebi.ac.uk/Tools/sss/
HMMER3 ³	[42]	http://hmmer.janelia.org/search

2.2 Alignment and Secondary Structure Prediction Tools

We find it best to run a variety of alignment and secondary structure prediction tools simultaneously. Each has its own peculiarities, and our hope is that by looking for shared homology and secondary structure predictions we can mitigate some of the problems discussed in the introduction. In this protocol, we use the WAR webserver[46] which allows the user to run 14 different methods simultaneously. These include Sankoff-type methods: FoldalignM[26], LocARNA[25], MXSCARNA[47], Murlet[48], and StrAL[49] + PETcofold[50]; Align-then-fold methods, which use a traditional alignment tool (ClustalW[51, 52] or MAFFT[53, 54]) followed by structure prediction (RNAalifold[29, 55] or Pfold[28]); Fold-then-align methods, which predict structures in all the input sequences and attempt to align these structures (RNAcast[56] + RNAforester[57]); Sampling methods which attempt to iteratively refine alignment and structure: MASTR[58] and RNASampler[59]; and other methods which do not fit in to the above traditional categories: CMfinder[60] and LaRA[61]. Finally, WAR also computes a consensus alignment using the alignments produced by all user-selected methods as input to the T-Coffee consistency-based aligner[32].

However, WAR is by no means exhaustive, and the applications may not be the most recent versions available. A number of groups maintain their own servers for RNA sequence analysis. Notable servers include the Vienna RNA WebServers[62], the Freiburg RNA Tools[63], the CBRC Functional RNA Project[64], and the Center for Non-Coding RNA in Technology and Health (RTH) Resources page. In addition, EMBL-EBI maintains a number of web-servers for popular multiple sequence alignment tools. Ultimately, as you become more comfortable with RNA sequence analysis you may want to begin installing and running new tools on a local *NIX machine; however, this is beyond the scope of the current chapter.

Resource	Reference	URL
WAR	[46]	http://genome.ku.dk/resources/war/
Vienna RNA	[62]	http://rna.tbi.univie.ac.at/
Freiburg RNA Tools	[63]	http://rna.informatik.uni-freiburg.de
CBRC Functional RNA Project	[64]	http://software.ncRNA.org
RTH Resources	NA	http://rth.dk/pages/resources.php
EMBL-EBI Alignment	NA	http://www.ebi.ac.uk/Tools/msa/

2.3 Genome Browsers

Genome browsers are essential for checking the context of putative homologs. The ENA[41] provides a no-frills sequence browser perfect for quickly checking annotations. For deeper annotations, the UCSC genome browser[65] and Ensembl[66] both contain a wide range of information for the organisms they cover. For bacterial and archaeal genomes, the Lowe lab maintains a modified version of the UCSC genome browser[67] which provides a number of tracks of particular interest to those working with ncRNA. The CBRC Functional RNA

³Currently amino acid only

Project maintains a UCSC genome browser mirror[64] for a number of eukaryotic organisms with a larger number of ncRNA-related tracks.

Resource	Reference	URL
European Nucleotide Archive	[41]	http://www.ebi.ac.uk/ena/
UCSC Genome Browser	[65]	http://genome.ucsc.edu/
Ensembl	[66]	http://www.ensembl.org
UCSC Microbial Genome Browser	[67]	http://microbes.ucsc.edu/
CBRC UCSC Genome Browser for Functional RNA	[64]	http://www.ncrna.org/glocal/cgi-bin/hgGateway

2.4 Alignment Editors

It is possible to edit alignments in any text editor; however we highly recommend using a secondary structure-aware editor such as Emacs with the RALEE major mode[68]. RALEE allows you to color bases according to base identity, secondary structure, or base conservation. It also allows the easy manipulation of sequences which are involved in structural interactions but are not close in sequence space through the use of split screens. A number of other specialized RNA editors are available: BoulderALE[69] and S2S[70] both allow the end user to visualize and manipulate tertiary structure in addition to secondary structure, and may be particularly useful if crystallographic information is available for your RNA. Other alternatives for editing RNA secondary structure are SARSE[71] and MultiSeq[72]. Recent versions of JalView[73] have begun to support RNA secondary structure as well, though this functionality isn't completely mature at the time of writing (late 2011.)

Resource	Reference	URL
RALEE	[68]	http://personalpages.manchester.ac.uk/staff/sam.griffiths-jones/software/ralee/
BoulderALE	[69]	http://www.microbio.me/boulderale
S2S	[70]	http://bioinformatics.org/S2S/
SARSE	[71]	http://sarse.ku.dk/
MultiSeq	[72]	http://www.ks.uiuc.edu/Research/vmd/plugins/multiseq/
JalView	[73]	http://www.jalview.org

2.5 Infernal

The centerpiece of our protocol is the Infernal package for constructing covariance models(CMs) from RNA multiple alignments[3]. We will use this to construct models of our RNA family. CMs model the conservation of positions in an alignment similar to a hidden Markov model(HMM), while also capturing *covariation* in structured regions[74, 75, 21]. Covariation is the process whereby a mutation of a single base in a hairpin structure will lead to selection in subsequent generations for compensatory mutations of its structural partner in

order to preserve canonical base-pairing, ie: Watson-Crick plus G-U pairs, and a functional structure. This combination of structural-evolutionary information has been shown to provide the most sensitive and specific homology search for RNA of any tools currently available[9, 76]. Unfortunately, this sensitivity and specificity come at a high computational cost, and Infernal searches can be time-consuming with genome-scale searches often taking hours on desktop computers. The development of heuristics to reduce this computational cost is an area of active research for the Infernal team, and has already been mitigated to some extent by the use of HMM filters and query-dependent banding of alignment matrices[77]. We refer the reader to Eric Nawrocki’s excellent primer on annotating functional RNAs in genomic sequence for a friendly introduction to the mechanics of the Infernal package[78].

Resource	Reference	URL
Infernal	[3, 78]	http://infernal.janelia.org/

3 Methods

We assume for the sake of this protocol that you are starting with a single sequence of interest. If you already have a set of putative homologs, you may wish to further diversify your collection of sequences using the methods described in section 3.1, or you may skip directly to section 3.2, or 3.4 if a secondary structure is known. No matter how many sequences you are starting with, it is always a good idea to run the tools available on the Rfam website (rfam.sanger.ac.uk) on them. This will verify that there isn’t already a CM available that covers your sequences. There are a number of other specialist databases that may also be worth searching if you have reason to believe your RNA sequence is a member of a well-defined class of RNAs, i.e. microRNAs, snoRNAs, rRNAs, tRNAs, etc. We have previously reviewed these databases in another book chapter[79]. A generic RNA sequence database aiming to capture all known RNA sequences, RNACentral[80] is currently in development and will provide a resource for easily identifying similar sequences with some evidence of transcription.

3.1 Gathering an initial set of homologous sequence

Now that you’ve confirmed that your sequence is novel, we will use NCBI-BLAST to identify additional homologous sequences. Once you’ve navigated to the nucleotide BLAST server there are a number of important options to set.

3.1.1 Setting NCBI-BLAST Parameters

First, it is important to choose a search set appropriate to your sequence. At this initial phase, we want to limit our exposure to sequences which are very distant from ours to avoid the number of obviously spurious alignments we will need to examine, increasing the power of our search. So, if your initial sequence is of human origin, you may want to limit your search to Mammalia, Tetrapoda, or Vertebrata depending on sequence conservation. Similarly, if you are working with an *Escherichia coli* sequence, you may want to limit your initial searches to Enterobacteriaceae or the Gammaproteobacteria. NCBI-BLAST searches are

relatively fast, so try several search sets to get a feel for how conserved your sequence is.

The second set of options to set is the “Program Selection” and the “Algorithm Parameters”. We recommend **blastn** as it allows for smaller word sizes. The word size describes the minimum length of an initial perfect match needed to trigger an alignment between our query sequence and a target. Smaller word sizes provide greater sensitivity, and seem to perform better for non-coding RNAs. We recommend a word size of 7, the smallest the NCBI-BLAST server allows.

Finally, you should set “Max Target Sequences” parameter to at least 1000. NCBI-BLAST returns hits in a ranked list from best match to worst by E-value (or the number of matches with the same quality expected to be found in a search over a database of this size), and will only display as many as “Max Target Sequences” is set to. We are primarily interested in matches on the edge of what NCBI-BLAST is capable of detecting reliably, and these will naturally fall towards the end of this list.

3.1.2 Selecting Sequences

Our goal at this stage is to pick a representative set of homologous sequences to “seed” our alignment with. As discussed in the introduction, single sequence alignment for nucleotides is generally only reliable to approximately 60 percent pair-wise sequence identity. At the same time, picking a large number of sequences with high percent identity can lead to *overfitting* of the secondary structure; that is, if our sequences are too similar we can end up predicting alignments and secondary structures which capture accidental features of a narrow clade, rather than the biologically relevant structure and sequence variation.

There are 3 major criteria we pick additional sequences based on, in rough order of importance: percent sequence identity, taxonomy, and sequence coverage. Handily, the NCBI-BLAST output displays measures of all of these. Our first selection criterion, percent identity, should fall between 65% and 95%; much lower and the sequence will be difficult to align, higher and it will be too similar to have any meaningful variation.

The second selection criterion, taxonomy, will depend somewhat on the organisms your sequence is associated with, but we generally want to limit the inclusion to a single (orthologous) instance per species. The exception to this rule is for diverged paralogous sequences within the species; if paralogs exist, you will need to decide how broadly you wish to define your family. Additionally, it may be useful to further limit the maximum percent identity to, say, 90% within a genus to further limit the number of highly similar sequences in your initial alignment.

Finally, assuming that you are sure of your sequence boundaries, we want to select sequences that cover the entire starting sequence. If you see many matches covering only a short section of your sequence, this may be due to the matching of a short convergent motif. This most commonly happens with the relatively long, highly-constrained bacterial rho-independent terminators, but may occur with other motifs. Alternatively, if you do not have well-defined sequence boundaries, you will need to determine these from the conservation you see in your BLAST hits – look for taxonomically diverse hits covering the same segment of your query sequence. In some cases, such as the long non-coding

RNAs, conserved domains may be much shorter than the complete transcribed sequence, but stay aware of the potential motif issue. A taxonomic distribution of sequences that makes biological sense given your knowledge of the molecule's function and that can be explained by direct inheritance of the sequence will be your best guide.

3.1.3 Examining Your Initial Homolog Set

Once you have assembled a set of sequences fitting the criteria described above, it is worth taking a closer look at them. Remember that these sequences will form the core of your alignment and CM, and errors at this stage can dramatically bias your results. A good first test is to examine the taxonomy of your sequences, and make sure it makes sense. Can you identify a clear pattern of inheritance that might explain the taxonomic distribution you see at this stage? Another good check is to examine your sequences in the ENA browser, or a domain-specific browser if one exists for your organisms. For many independently transcribed RNAs, genomic context is more conserved than sequence, and ncRNA genes will often fall in homologous intergenic or intronic regions even at large evolutionary distances. If you are particularly ambitious, and the tools are available for your organisms of interest, you may wish to try to identify promoter sequence upstream of your candidate or terminator sequence downstream. If your sequence is a putative cis-regulatory element, such as a riboswitch, thermosensor, or attenuator, you may want to check that it occurs upstream of genes with similar functions or in similar pathways. Finally, it is always worth searching your putative homologs through the Rfam website even if your initial sequence had no matches – Rfam's models are not perfect, and may miss distant homologs of known families.

3.2 Aligning and predicting secondary structure

We will use the WAR servers to construct an initial alignment. Because of the criteria we've set for sequence similarity in our gathering step, all of the sequences in our initial homolog set should have at least 60% pairwise sequence identity with at least one other sequence in the set. Under these conditions sequence-only alignment methods using primary sequence information only can perform adequately, as discussed previously. These methods combined with alignment folding tools which identify for conserved structural signals and covariation can produce reasonable predicted secondary structures[10]. However it is still often useful to observe the behavior of as many alignment tools as possible. Using WAR, for a fairly fast alignment we recommend running CMfinder[60], StrAL+PETfold[49, 50], ClustalW[51, 52] and MAFFT[53, 54] with RNAalifold[29, 55] and Pfold[28]. WAR will also produce a consensus alignment using T-Coffee[32], which will attempt to find an alignment consistent with all of the individual alignments produced by other methods.

Once WAR returns your alignment results, there are a number of things you should take a note of that will assist you in picking an alignment and further in manual refinement. First, the consensus alignment page will display a graphical representation of the consistency of the alignments which will allow you to quickly tell which areas of the alignment may require attention during manual refinement, or areas that may harbor structure not captured by the

T-coffee consensus alignment

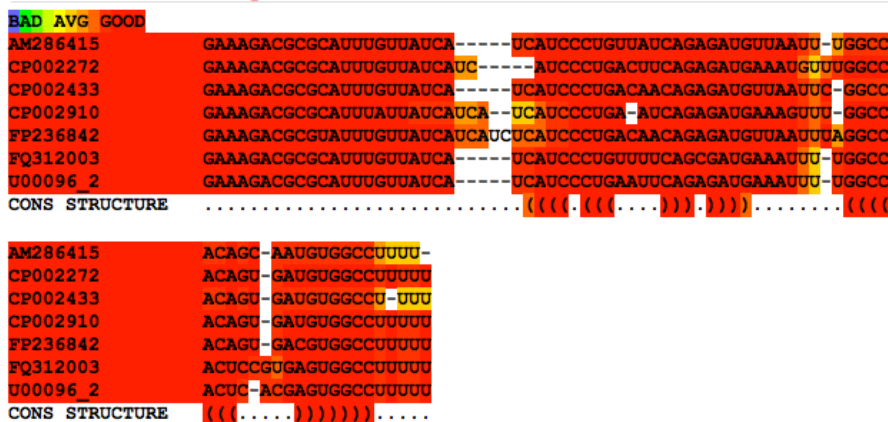


Figure 1: T-coffee consensus alignment for close MicA homologs produced by WAR, colored for alignment consistency between methods. Due to the high percent identity in these sequence, the alignments are highly consistent, though even here the areas of lower consistency are informative for manual refinement - see section 4.

majority consensus. The consensus can be recomputed based on differing subsets alignment methods, if you believe one method (or set of methods) may be unduly influencing the consensus. Once you’ve carefully looked over the consensus alignment, examine each alignment produced by WAR in turn: What structures are shared? Where do the alignments differ from each other? Can you identify any sequence or structural motifs which may help to guide your alignment? At this level of sequence identity, you should hope to see fairly consistent alignments in functional regions of the sequence, interspersed with more difficult to align regions, presumably under less severe selective pressure. Often the consensus alignment is a good choice to move forward with. However, there are cases where certain classes of tools will obviously mis-align regions of the sequence and bias the consensus. Keep in mind what you’ve seen in the alternative alignments as well; this information may be useful in manual refinement. You will want to save the stockholm file for the alignment you’ve chosen to your local computer at this point.

Later in the family-building process when you have identified more distant homologs, the average pair-wise identity of the sequences in your data set may have dropped below 60%. At this point, you may want to begin including some of the Sankoff-type alignment methods available in WAR. Using these methods can dramatically increase the runtime for your sequence alignment jobs, though, particularly for sequences over a couple of hundred of bases long. We will discuss alternatives to re-aligning sequences during the iterative expansion of the alignment in section 3.5.

3.3 Manually refining alignments

Our goal in manual refinement is to attempt to correct errors made by automatic alignment tools. We generally use RALEE[68], an RNA editing mode for Emacs, for editing alignments. However, any editor you are comfortable with in which you can easily visualize sequence and structural conservation will work;

a number of alternative editors are listed in the Materials section.

A good place to start editing is around the edges of predicted hairpin structures. Are there base-pairs which appear to be misaligned? Can you add base-pairs to the structure? Are there predicted base-pairs which don't appear to be well conserved that should be trimmed? Can individual bases be moved in the alignment to create more convincing support for the predicted structure?

Once you are satisfied with your manual refinement of predicted secondary structure elements, next you should turn your attention to areas identified as uncertain in the WAR/T-Coffee consensus alignment. Were there alternative structures predicted in these regions? Do you see support for these structures in the sequences? If these regions are unstructured, can you identify any conserved sequence motifs in the region? If you will be regularly working with a particular class of ncRNA, it can be useful to familiarize yourself with predicted binding motifs of associated RNA-binding proteins as these are likely to be conserved but can have many variable positions.

At this stage, it is also possible to include information from experimental data. Crystal structure information from a single sequence in the SEED alignment can be used to validate and improve a predicted secondary structure. Tertiary structure-aware editors such as BoulderAle[69] can help in applying this information to the alignment. Other experimental evidence, such as chemical footprinting can also provide valuable information. Knowing whether even a single base is involved in a pairing interaction can drastically reduce the space of possible structures the sequence can fold in to, simplifying the problem of predicting secondary structure. Both the RNAfold and RNAalifold web servers available through the Vienna RNA website[62] are capable of taking advantage of this information in the form of folding constraints. We hope that these sorts of datasets will become widely available in consistent formats in the near future[81].

3.4 Building a covariance model

For those comfortable with the *NIX command line, building an Infernal CM is fairly straight-forward. We refer the reader to the User's Guide available from the Infernal website (<http://infernal.janelia.org>) for installation instructions and a detailed tutorial. The basic syntax to build and calibrate a family is:

```
> cmbuild my.cm my.sto
> cmcalibrate my.cm
```

The first command constructs the CM (`my.cm`) from the alignment you've carefully curated (`my.sto`). The second command calibrates the various filters Infernal uses to accelerate its search using simulated sequences generated from the CM. Note that calibration can take a long time – hours for longer models. You can get a quick estimate of the time calibration will take using the command:

```
> cmcalibrate --forecast 1 my.cm
```

Congratulations! You should now have a working CM for your RNA family. This is a fully capable model, and can be used as is for homology search and genome annotation. However, as it stands, your CM will only capture the

sequence diversity which was able to be detected by our initial BLAST search. In order to fully take advantage of the power of CMs, it is necessary to expand the diversity of the sequence it is trained on through iterative expansion of our initial set of sequence homologs.

3.5 Strategies for expanding model coverage

3.5.1 Plan A: Iterative search of sequence databases

The method Rfam uses to identify more divergent homologs to seed sequences is to pre-filter CM-based searches with sequence-based homology search tools. This allows us to cover a large sequence space with a (comparatively) modest investment of computational time. Any of the single sequence search tools mentioned in section 2.1 would make an effective pre-filter.

The easiest way to preform filtering yourself is to use the NCBI BLAST webserver to search each sequence in your seed alignment following the methods outlined for collecting your initial set of homologs in section 3.1. You may wish to relax the criteria slightly, then use the CM to preform a more sensitive search on this set of filtered sequences. This will enable you to detect more distantly related sequences, though you should always examine sequence context and the phylogenetic relationship between sequences as a sanity check before including them in your seed. These methods can be automated with basic scripting and bioinformatics modules such as BioPerl[82] or Biopython[83], though this is beyond the scope of this chapter.

Once you have identified a new set of homologs, you can align them to your previous CM using Inferal's `calign`:

```
> calign my.cm newsequences.fasta > newsequences.sto
```

This alignment can then be merged with your original alignment:

```
> calign --merge my.cm my.sto newsequences.sto > combined_alignment.sto
```

This alignment can then be used to build a new CM, which will capture the additional sequence variation you have discovered in your BLAST searches.

The disadvantage of this method is that each search only uses the information available in a single sequence, meaning that valuable information about variation is lost and as a result the power of the search suffers. Fast profile-based methods such as HMMER3[2] will hopefully remedy this problem in the near future, but these methods are not mature for DNA and RNA sequence at the present. Older versions of HMMER can be used to search DNA sequence with increased power, but they require more computational resources than BLAST (though far less than Inferal) and need to be used at the command-line.

3.5.2 Plan B: Directed search of chosen sequences

Another approach is to run the unfiltered CM over selected genomes or genomic regions. While the greater sensitivity and specificity of this method can help identify more distant homologs than is possible with BLAST, it has the disadvantage that it requires a much larger investment of computational resources to provide an equivalent phylogenetic coverage. This method can be particularly powerful in bacterial and archaeal genomes, where small genome size allows us

to search a phylogenetically-representative sample of genomes in less than a day on a desktop computer. In the case of larger eukaryotic genomes, it may be necessary to search a few genomes to determine if homologs of your RNA are likely to exist in certain lineages, then extract homologous intergenic regions to continue searching. Our rationale here is much the same as in limiting the database for our initial BLAST search: by only looking in genomes where we have some prior belief that they may contain homologous sequence we reduce the noise in our low-scoring hits, meaning that we have to manually examine less hits to establish a score threshold for likely homologs.

Once you have examined candidates following the principles outlined earlier, it is easy to incorporate your new sequences using the *easel* package included with *Infernal*. First, search the genome generating a tabfile:

```
> cmsearch --tabfile searchfile.tab my.cm genome.fasta
```

Then use *easel* to index the genome and extract the hits:

```
> esl-sfetch --index genome.fasta
> esl-sfetch --tabfile genome.fasta searchfile.tab > hits.fasta
```

These sequences can then be aligned and merged as with BLAST hits. Alternatively, if you discover a divergent lineage, it may be easiest to construct a separate alignment for these sequences, then use shared structural and sequence motifs to manually combine the two alignments. Sankoff-type alignment method may also be useful for aligning divergent clades.

3.5.3 Plan C: When A and B fail...

In some cases, it will be very difficult to identify homologs of a candidate RNA across its full phylogenetic range. This can be because of high sequence variability, as in the Vault RNAs[84]. Alternatively, some longer RNAs, such as the RNA component of the telomerase ribonucleoprotein, consist of well-conserved segments interspersed with long variable regions which can't be easily discovered by standard search with naive covariance models.

A number of computational techniques exist for approaching these difficult cases, reviewed by Mosig and colleagues[85]. These methods include *fragrep2*[86], which allows the user to search fragmented conserved regions, *fragrep3*, which allows the user to incorporate custom structural motifs with fragmented search, and *GotohScan*[87], which implements a *semi-global* alignment algorithm that will align a query sequence to a (potentially) extended genomic region.

4 An example: MicA

We will now illustrate some of the concepts we've discussed using the example of MicA, an Hfq-dependent bacterial trans-acting antisense small RNA (sRNA). Many bacterial sRNAs are similar in function to eukaryotic microRNAs, pairing to target mRNA transcripts through a short antisense-binding region, generally targeting the transcript for degradation[88]. MicA is known to target a wide-range of outer membrane protein mRNAs using a 5' binding-region in both *E. coli*[89] and *S. enterica*[90] in response to membrane stress. The current covariance model for MicA (accession RF00078) in Rfam (release 10.1) is largely

restricted to *E. coli*, *S. enterica*, and *Y. pestis*. Here, as an example, we will attempt to improve on this model using the methods we’ve described in this chapter. In the process, we discover previously unreported homologs in the nematode symbionts of the Gammaproteobacterial genus *Xenorhabdus*.

For our starting point, we are using the MicA sequence from Gisela Storz’s spreadsheet of known *E. coli* sRNAs[91]:

MicA: GAAAGACGCGCATTGTTATCATCATCCCTGAATTCAGAGATGAAATTTGGCCACTCACGAGTGGCCTTTTT

It is a useful exercise to compare the single sequence predicted secondary structures for this sequence and the *E. coli* sequence from the current Rfam SEED alignment(see Figure 2). This illustrates that even for nearly identical sequences, single sequence structure prediction methods can give divergent results. Other important features to notice are that the 3’ hairpin shared by the predicted structures appears to be a rho-independent terminator, and this could be confirmed with a motif hunting tool[37] and used during manual curation.

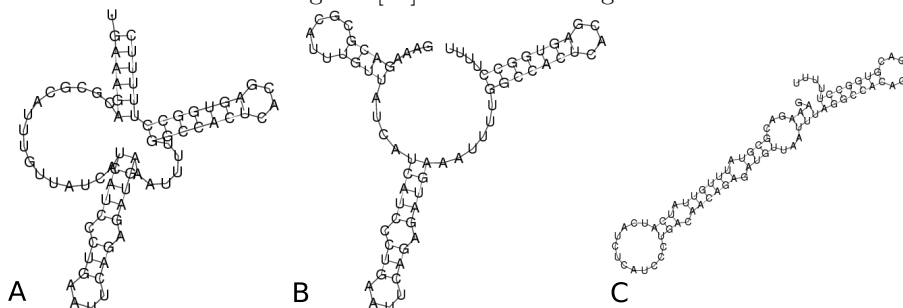


Figure 2: Alternative structures predicted by the RNAfold webserver for single MicA sequences. A) *E. coli* APEC sequence from the current Rfam seed alignment. B) *E. coli* sequence from Storz’s sRNA spreadsheet. C) A likely homolog from *Erwinia pyrifoliae*. Notice the differences in the secondary structure of the first two examples, despite only differing by two extra nucleotides at the gene boundaries. The *Erwinia* prediction only shares a single stem with the *E. coli* predictions, despite relatively high sequence similarity.

We now begin by following the guidance in section 3.1 to collect an initial set of putative homologs. To obtain an initial set of sequences, we BLAST the *E. coli* MicA sequence over the nucleotide collection database limited to the enterobacteria (taxonomy id: 543) using the blastn algorithm. The BLAST search returns a number of highly similar *E. coli* sequences, as well as related sequences from the closely related *S. enterica*. As we move down to less similar sequences (as judged by their E-values) we identify progressively more evolutionarily distant organisms.

From these sequences, we want to select a group of sequences with a reasonably diverse taxonomic range and as much sequence diversity as possible, while being reasonably confident that they are true homologs. In this case we will choose based on maximizing genus diversity, a percent id between 75% and 90%, and 100% sequence coverage as we’re fairly confident in the MicA gene boundaries. For our initial alignment, we have chosen sequences from *Salmonella typhimurium* (EMBL-Bank accession: FQ312003), *Klebsiella pneumoniae* (CP002910), *Enterobacter cloaca* (CP002272), *Yersinia pestis* (AM286415), *Pantoea* sp. At-9b (CP002433), and *Erwinia pyrifoliae* (FP236842). From a quick examination with the ENA browser, it appears that all of these sequences fall in a intergenic region between a luxS protein homolog and a gshA protein homolog, further increasing our confidence that these are true homologs. From our results, we can also see a few promising hits that don’t quite meet our crite-

CP000308.1	Yersinia pestis Antiqua, complete genome	68.0	68.0	100%	1e-10	80%
CP000305.1	Yersinia pestis Nepal516, complete genome	68.0	68.0	100%	1e-10	80%
AE009952.1	Yersinia pestis KIM, complete genome	68.0	68.0	100%	1e-10	80%
BX936398.1	Yersinia pseudotuberculosis IP32953 genome, complete sequence	68.0	68.0	100%	1e-10	80%
CP002038.1	Dickeya dadantii 3937, complete genome	62.6	94.5	100%	5e-09	81%
CP002154.1	Edwardsiella tarda FL6-60, complete genome	62.6	62.6	77%	5e-09	85%
CP001135.1	Edwardsiella tarda EIB202, complete genome	62.6	62.6	77%	5e-09	85%
EU070919.1	Edwardsiella tarda strain TX1 autoinducer-2 synthase (luxS) gene, complete sequence	62.6	62.6	77%	5e-09	85%
CP001836.1	Dickeya dadantii Ech586, complete genome	60.8	60.8	77%	2e-08	83%
FR719187.1	Erwinia amylovora ATCC BAA-2158, whole genome shotgun sequence	59.0	59.0	100%	7e-08	78%
CP002124.1	Erwinia sp. Ejp617, complete genome	59.0	59.0	100%	7e-08	78%
FN434113.1	Erwinia amylovora CFBP1430 complete genome	59.0	59.0	100%	7e-08	78%
FN666575.1	Erwinia amylovora ATCC 49946 chromosomal sequence	59.0	59.0	100%	7e-08	78%
FN392235.1	Erwinia pyrifoliae DSM 12163 complete genome, culture collection DSI	59.0	59.0	100%	7e-08	78%
CP001790.1	Pectobacterium wasabiae WPP163, complete genome	59.0	59.0	77%	7e-08	85%
FP236842.1	Erwinia pyrifoliae strain Ep1/96 complete chromosome	59.0	59.0	100%	7e-08	78%
CP001657.1	Pectobacterium carotovorum subsp. carotovorum PC1, complete genome	59.0	90.9	77%	7e-08	88%
AJ628151.1	Erwinia carotovora subsp. carotovora luxS gene for autoinducer-2 synthase	59.0	59.0	77%	7e-08	85%
BX950851.1	Erwinia carotovora subsp. atroseptica SCRI1043, complete genome	59.0	59.0	77%	7e-08	85%
CP001655.1	Dickeya zeae Ech1591, complete genome	57.2	57.2	77%	2e-07	82%
CU468135.1	Erwinia tasmaniensis strain ET1/99 complete chromosome	51.8	51.8	100%	1e-05	75%
AP008232.1	Sodalis glossinidius str. 'morsitans' DNA, complete genome	51.8	51.8	43%	1e-05	96%
AJ628152.1	Serratia sp. luxS gene for autoinducer-2 synthase, strain ATCC 39006	51.8	51.8	48%	1e-05	94%
AC192956.2	Candidatus Regiella insecticola clone CUGI_APP_BA-03-N07, complete genome	48.2	48.2	97%	1e-04	75%
CP001600.1	Edwardsiella ictaluri 93-146, complete genome	46.4	81.9	83%	4e-04	86%
FN545200.1	Arsenophonus nasoniae whole genome shotgun assembly, contig scaff	33.7	33.7	97%	2.7	70%
FR775239.1	Salmonella enterica subsp. enterica serovar Weltevreden str. 2007-60	31.9	31.9	37%	9.3	85%
CP001895.1	Pantoea vagans C9-1 plasmid pPag3, complete sequence	31.9	31.9	41%	9.3	83%
FN667742.1	Xenorhabdus nematophila ATCC 19061 chromosome, complete genome	31.9	31.9	27%	9.3	95%
FN667741.1	Xenorhabdus bovienii SS-2004 chromosome, complete genome	31.9	95.8	73%	9.3	100%
FM162591.1	Photorhabdus asymbiotica ATCC43949 complete genome	31.9	31.9	27%	9.3	95%
BA000021.3	Wigglesworthia glossinidia endosymbiont of Glossina brevipalpis DNA,	31.9	31.9	30%	9.3	90%
AL627281.1	Salmonella enterica serovar Typhi (Salmonella typhi) strain CT18, complete genome	31.9	31.9	37%	9.3	85%
X79787.1	E.coli nrdEF operon (partial)	31.9	31.9	30%	9.3	90%

Figure 3: Truncated results from a NCBI-BLAST search of the *E. coli* MicA sequence, showing the low E-value results. We are primarily interested in column 2 for genus and species information, column 5 for sequence coverage information, and column 7 for percent identity informations.

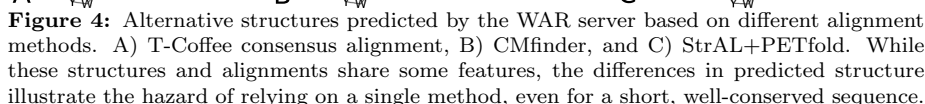
ria, such as *Dickeya*, *Xenorhabdus*, *Photorhabdus* and *Wigglesworthia*. We will keep these in mind later to expand our coverage.

Now that we have a starting set of sequences, we can assemble them in to a fasta file:

```
>U00096.2
GAAAGACGCGCATTGTTATCATCATCCCTGAATTCAGAGATGAAATTTTGGCCACTCACGAGTGGCCTTTTT
>FQ312003
GAAAGACGCGCATTGTTATCATCATCCCTGTTTTCAGCGATGAAATTTTGGCCACTCCGTGAGTGGCCTTTTT
>CP002272
GAAAGACGCGCATTGTTATCATCATCCCTGACTTCAGAGATGAAATGTTTGGCCACAGTGATGTGGCCTTTTT
>CP002910
GAAAGACGCGCATTATTATCATCATCATCCCTGAATCAGAGATGAAAGTTTGGCCACAGTGATGTGGCCTTTTT
>AM286415
GAAAGACGCGCATTGTTATCATCATCCCTGTTATCAGAGATGTTAATTTGGCCACAGCAATGTGGCCTTTTT
>CP002433
GAAAGACGCGCATTGTTATCATCATCCCTGACAACAGAGATGTTAATTCGGCCACAGTGATGTGGCCTTTTT
>FP236842
GAAAGACGCGTATTGTTATCATCATCTCATCCCTGACAACAGAGATGTTAATTTAGGCCACAGTGACGTGGCCTTTTT
```

We can use this to run WAR, and look at the secondary structures predicted by each method. One secondary structure appears to dominates the predictions. However, its important to check the other predicted secondary structures - do any of them pick up convincing substructures that may have been missed by other methods?

In this case, the consensus alignment (see Figure 1) seems to agree well with the majority of alignment and structure prediction methods, and is consistent with previous experimental probing[92]. We can improve the alignment manually. The first basepair in the first stem in CP002433 can be rescued by shifting



```
# STOCKHOLM 1.0

#=GF AU WAR

RM286415
CP002272
FP236842
QF312003
UO096_2
#=GC SS_cons
#=GC RF
//
# STOCKHOLM 1.0

#=GF AU WAR

RM286415
CP002272
FP236842
QF312003
UO096_2
#=GC SS_cons
#=GC RF
//
# STOCKHOLM 1.0

#=GF AU WAR

RM286415
CP002272
FP236842
QF312003
UO096_2
#=GC SS_cons
#=GC RF
//
# STOCKHOLM 1.0

#=GF AU WAR

RM286415
CP002272
FP236842
QF312003
UO096_2
#=GC SS_cons
#=GC RF
//
# STOCKHOLM 1.0

#=GF AU WAR

RM286415
CP002272
FP236842
QF312003
UO096_2
#=GC SS_cons
#=GC RF
//
```

Now we will follow Plan B to add sequences to our alignment using the genomes for the low-scoring BLAST hits we had previously made a note of while collecting our initial set of sequences, though you could also choose these sequences based on your knowledge of your organisms phylogeny or the suspected function of your RNA. The genomes we've chosen here are *Dickeya*


```

# STOCKHOLM 1.0
#=GF AU WAR
FN545200 GCAAGACGCG-AAAAUUGUUAUCAUC-CUAUUCUUAGA.GAUUUUUUUUGGCCACUUUAAGUGGCCAUUUU
FN667741 G-AAGACGCGCAAAAUUGUUAUCAUCCUAUUUUUUAGA.GAUUUUUUUUGGCCACU-GGUGGCCAUUUU
FN667742 G-AAGACGCGCAAAAUUGUUAUUAUCCUAUUUCUUAGAACUU-UUUU-GGCCAC-CUC-GUGGCCAUUUU
#=GC SS_cons .....<<<<<<.....>>>>>>.....<((((((.....))))))>.....
#=GC RF GcAAGACGCGCAAAAUUGUUAUCAUCCUAUUUCUUAGAAGAYUUUUUUUUGGCCACCMdMRGUGGCCAUUUU
//

```

Figure 8: An alignment of *Xenorhabdus* homologs.

```

# STOCKHOLM 1.0
U00096.2 GAAAGACGCGCA..UUUGUUAUCA...UCAUCCUGAAUU.CAGAGAUG.....AAAU,UUU,GGCCACU..CA,CG....AGUGGCC..UUUUU
F0312003 GAAAGACGCGCA..UUUGUUAUCA...UCAUCCUGUUUU.CAGAGAUG.....AAAU,UUU,GGCCACU..CCgUG....AGUGGCC..UUUUU
CP002272 GAAAGACGCGCA..UUUGUUAUCA...UCAUCCUGACUU.CAGAGAUG.....AAAUgUUU,GGCCACA..GU,GA....UGUGGCC..UUUUU
CP002910 GAAAGACGCGCA..UUUAUUUAUCAuca..UCAUCCUGA-AU.CAGAGAUG.....AAA,GUUU,GGCCACA..GU,GA....UGUGGCC..UUUUU
FM286415 GAAAGACGCGCA..UUUGUUAUCA...UCAUCCUGUUUU.CAGAGAUG..UU..AA...UUU,GGCCACA..GC,AA....UGUGGCC..UU-UU
CP002433 GAAAGACGCGCA..UUUGUUAUCA...UCAUCCUGACAA.CAGAGAUG..UU..AA...UUCGGCCACA..GU,GA....UGUGGCC..UU-UU
FP236842 GAAAGACGCGA...UUUGUUAUCucauUCAUCCUGACAA.CAGAGAUG..UU..AA...UUUAGGCCACA..GU,GA....UGUGGCC..UUUUU
AF008232 GAAAGACGCGCA..UUUGUUAUCA...UCAUCCUGUUA.CAGAGAUG..UU..AA...UUUA,GGCCACA..GUuUC....UGUGGCC..UUU-UU
CP001655 GAAAGACGCGCA..UUUAUUUAUCA...UCAUCCUGUUUAU.AGAGAUG..UU...UCUUUC,GGCCACgGUAAcaucgGGUGGC..AUU-UU
FN545200 GCAAGACGCG-AAAA,UUGUUU...CAUCC-UAUUCUUA.AGA.GAUUUA..U,UUU,GGCCACU..UUAA....AGUGGCCAUU-UU
FN667741 G-AAGACGCGCAAAA,UUGUUU...CAUCCUAUUUUUA.AGA.GAUUUA...UUUCGGCCACU..ACU-....GGUGGCCAUU-UU
FN667742 G-AAGACGCGCAAAA,UUGUUU...UAUCCUAUUUCUA.AGAUACUU-UA...UUCGGCCAG...CUC-....GUGGCCAUU-UU
#=GC SS_cons .....<<<<<<.....>>>>>>.....<<<<<<.....>>>>>>.....
//

```

Figure 9: Divergent *Xenorhabdus* homologs manually merged with the MicA alignment. Notice the variation in both secondary structure and sequence conservation added by these sequences.

References

- [1] SF Altschul, W Gish, W Miller, EW Myers, and DJ Lipman. Basic local alignment search tool. *J Mol Biol.*, 215(3):403–410, Oct 1990.
- [2] SR Eddy. Accelerated profile HMM searches. *PLoS Comput Biol.*, in press, 2011.
- [3] EP Nawrocki, DL Kolbe, and SR Eddy. Infernal 1.0: Inference of RNA alignments. *Bioinformatics*, 25:1335–1337, 2009.
- [4] RD Finn, J Mistry, J Tate, P Coghill, A Heger, JE Pollington, OL Gavin, P Gunesekaran, G Ceric, K Forslund, L Holm, EL Sonnhammer, SR Eddy, and A Bateman. The Pfam protein families database. *Nucl Acids Res.*, 38(Database issue):D211–D222, 2010.
- [5] PP Gardner, J Daub, J Tate, BL Moore, IH Osuch, S Griffiths-Jones, RD Finn, EP Nawrocki, DL Kolbe, SR Eddy, and A Bateman. Rfam: Wikipedia, clans and the “decimal” release. *Nucl Acids Res.*, 39(Database issue):D141–D145, Nov 2011.
- [6] B Rost. Twilight zone of protein sequence alignments. *Protein Eng.*, 12(2):85–94, Oct 1999.
- [7] JD Thompson, F Plewniak, and O Poch. A comprehensive comparison of multiple sequence alignment programs. *Nucl Acids Res.*, 27(13):2682–2690, May 1999.
- [8] PP Gardner, A Wilm, and S Washietl. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucl. Acids Res.*, 33(8):2433–2439, Apr 2005.
- [9] EK Freyhult, JP Bollback, and PP Gardner. Exploring genomic dark matter: A critical assessment of the performance of homology search methods on noncoding RNA. *Genome Res.*, 17(1):117–125, Jan 2007.

- [10] PP Gardner and R Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5(140), Sep 2004.
- [11] GG Brownlee. Sequence of 6S RNA of E. coli. *Nat New Biol.*, 229(5):147–149, Feb 1971.
- [12] KM Wassarman and G Storz. 6S RNA regulates E. coli RNA polymerase activity. *Cell*, 101(6):613–623, Jun 2000.
- [13] JE Barrick, N Sudarsan, Z Weinberg, WL Ruzzo, and RR Breaker. 6S RNA is a widespread regulator of eubacterial RNA polymerase that resembles an open promoter. *RNA*, 11(5):774–784, May 2005.
- [14] CM Sharma, S Hoffman, F Darfeuille, J Reignier, S Findeiss, A Sittka, S Chabas, K Reiche, J Hackermüller, R Reinhardt, PF Stadler, and J Vogel. The primary transcriptome of the major human pathogen *helicobacter pylori*. *Nature*, 464(7286):250–255, Mar 2010.
- [15] Z Weinberg, JX Wang, J Bogue, J Yang, K Corbino, RH Moy, and RR Breaker. Comparative genomics reveals 104 candidate structured RNAs from bacteria, archaea, and their metagenomes. *Genome Biol.*, 11(3):R31, Mar 2010.
- [16] P Menzel, J Gorodkin, and PF Stadler. The tedious task of finding homologous noncoding RNA genes. *RNA*, 15:2075–2082, Oct 2009.
- [17] PP Gardner and A Bateman. A home for RNA families at RNA Biology. *RNA Biology*, 6(1):2–4, Jan 2009.
- [18] D Sankoff. Simultaneous solution of the RNA folding, alignment and proto-sequence problems. *SIAM J Appl Math.*, 45(5):810–825, Oct 1985.
- [19] SR Eddy. What is dynamic programming? *Nat Biotechnol.*, 22(7):909–910, Jul 2004.
- [20] SR Eddy. How do RNA folding algorithms work? *Nat Biotechnol.*, 22(11):1457–1458, Nov 2004.
- [21] R Durbin, S Eddy, A Krogh, and G Mitchison. *Biological sequence analysis*. Cambridge University Press, 1998.
- [22] SB Needleman and CD Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol.*, 48(3):443–453, Mar 1970.
- [23] R Nussinov, G Pieczenik, JR Griggs, and DJ Kleitman. Algorithms for loop matchings. *SIAM J Appl Math.*, 35(1), Jul 1978.
- [24] M Hamada, K Sato, K Hisanori, T Mituyama, and K Asai. Centroidalign: fast and accurate aligner for structured RNAs by maximizing expected sum-of-pairs score. *Nucl Acids Res.*, 25(24):3236–3243, Sep 2009.
- [25] S Will, K Reiche, IL Hofacker, PF Stadler, and R Backofen. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput Biol.*, 3(4):e65, Apr 2007.

- [26] E Torarinsson, JH Havgaard, and J Gorodkin. Multiple structural alignment and clustering of RNA sequences. *Bioinformatics*, 23(8):926–932, Feb 2007.
- [27] RK Bradley, L Pachter, and I Holmes. Specific alignment of structured RNA: stochastic grammars and sequence annealing. *Bioinformatics*, 24(23):2677–2683, Sep 2008.
- [28] B Knudsen and Hein J. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucl Acids Res.*, 31(13):3423–3428, Apr 2003.
- [29] SH Berhart, IL Hofacker, S Will, AR Gruber, and PF Stadler. RNAali-fold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, 9:474, Nov 2008.
- [30] M Hamada, H Kiryu, K Sato, and K Mituyama, T amd Asai. Predictions of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, 25(4):465–473, Nov 2009.
- [31] A Löytynoja and N Goldman. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. *Science*, 320:1632–1635, Jun 2008.
- [32] C Notredame, DG Higgins, and J Heringa. T-coffee: a novel method for fast and accurate multiple sequence alignment. *J Mol Biol.*, 302(1):205–217, Sep 2000.
- [33] AS Schwartz and L Pachter. Multiple alignment by sequence annealing. *Bioinformatics*, 23(ECCB 2006):e24–e39, Jan 2006.
- [34] AS Schwartz, E Myers, and L Pachter. Alignment metric accuracy. *arXiv:q-bio.QM/0510052*, 2006.
- [35] RK Bradley, A Roberts, M Smoot, S Juvekar, J Do, C Dewey, I Holmes, and L Pachter. Fast statistical alignment. *PLoS Comput Biol.*, 5(5), May 2009.
- [36] KM Wong, MA Suchard, and JP Huelsenbeck. Alignment uncertainty and genomic analysis. *Science*, 319:473–476, Jan 2008.
- [37] PP Gardner, L Barquist, A Bateman, EP Nawrocki, and Z Weinberg. RNIE: genome-wide prediction of bacterial intrinsic terminators. *Nucl Acids Res.*, 39(14):5845–5852, Apr 2011.
- [38] J Livny, MA Fogel, BM Davis, and MK Waldor. sRNAPredict: an integrative computational approach to identify sRNAs in bacterial genomes. *Nucl Acids Res.*, 33(13):4096–4105, Jun 2005.
- [39] PP Gardner, J Daub, JG Tate, EP Nawrocki, DL Kolbe, S Lindgreen, AC Wilkinson, RD Finn, S Griffiths-Jones, SR Eddy, and A Bateman. Rfam: updates to the RNA families database. *Nucl Acids Res.*, 37(suppl 1):D136–D140, Oct 2009.

- [40] M Johnson, I Zaretskaya, Y Raytselis, S McGinnis, and TL Madden. NCBI BLAST: a better web interface. *Nucl Acids Res.*, 36(suppl 2):W5–W9, Apr 2008.
- [41] R Leinonen, R Akhtar, E Birney, L Bower, A Cerdeno-Tárraga, Y Cheng, I Cleland, N Faruque, N Goodgame, R Gibson, G Hoad, M Jang, N Pakseresht, S Plaister, R Radhakrishnan, K Reddy, S Sobhany, PT Hoopen, R Vaughn, V Zalunin, and G Cochrane. The European Nucleotide Archive. *Nucl Acids Res.*, 39(suppl1):D28–D31, Oct 2010.
- [42] RD Finn, J Clements, and SR Eddy. HMMER web server: interactive sequence similarity searching. *Nucl Acids Res.*, in press, 2011.
- [43] E Myslinski, V Ségault, and C Branlant. An intron in the genes for U3 small nucleolar RNAs of the yeast *Saccharomyces cerevisiae*. *Science*, 247(4947):1213–6, Mar 1990.
- [44] Kent WJ. BLAT—the BLAST-like alignment tool. *Genome Res.*, 12(4):656–664, Apr 2002.
- [45] E Birney, M Clamp, and R Durbin. Genewise and genomewise. *Genome Res*, 14(5):988–95, May 2004.
- [46] E Torarinsson and S Lindgreen. WAR: Webserver for aligning structural RNAs. *Nucl Acids Res.*, 36(suppl 2):W79–W84, May 2008.
- [47] Y Tabei, H Kiryu, T Kin, and K Asai. A fast structural alignment method for long RNA sequences. *BMC Bioinformatics*, 9(33), Jan 2008.
- [48] H Kiryu, Y Tabei, T Kin, and K Asai. Murlet: a practical multiple alignment tool for structural RNA sequences. *Bioinformatics*, 23(13):1588–1598, Apr 2007.
- [49] D Dalli, A Wilm, I Mainz, and G Steger. StrAl: progressive alignment of non-coding RNA using base pairing probability vectors in quadratic time. *Bioinformatics*, 22(13):1593–1599, Apr 2006.
- [50] SE Seemann, J Gorodkin, and R Backofen. Unifying evolutionary and thermodynamic information for RNA folding of multiple alignments. *Nucl Acids Res.*, 36(20):6355–6362, Aug 2008.
- [51] JD Thompson, DG Higgins, and TJ Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl Acids Res.*, 22(22):4673–4680, 1994.
- [52] R Chenna, H Sugawara, T Koike, R Lopez, T J Gibson, D G Higgins, and J D Thompson. Multiple sequence alignment with the clustal series of programs. *Nucleic Acids Res*, 31(13):3497–3500, Jul 2003.
- [53] K Katoh, G Asimenos, and H Toh. Multiple alignment of DNA sequences with MAFFT. *Methods Mol Biol*, 537:39–64, 2009.

- [54] K Katoh, K Misawa, K Kuma, and T Miyata. MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform/. *Nucl Acids Res.*, 30:3059–3066, 2002.
- [55] I L Hofacker. RNA consensus structure prediction with RNAalifold. *Methods Mol Biol*, 395:527–44, 2007.
- [56] J Reeder and R Giegerich. Consensus shapes: An alternative to the Sankoff algorithm for RNA consensus structure prediction. *Bioinformatics*, 21(17):3516–3523, 2005.
- [57] M Höchsmann, T Töller, R Giegerich, and S Kurtz. Local similarity of RNA secondary structures. *Proc. of the IEEE Bioinformatics Conference*, pages 159–168, 2003.
- [58] S Lindgreen, P Gardner, and A Krogh. MASTR: Multiple alignment and structure prediction of non-coding RNAs using simulated annealing. *Bioinformatics*, 23(24):3304–3311, Nov 2007.
- [59] Xu. X, Y Ji, and G Stormo. RNA Sampler: A new sampling based algorithm for common RNA secondary structure prediction and structural alignment. *Bioinformatics*, 23(15):1883–1891, 2007.
- [60] Z Yao, Z Weinberg, and WL Ruzzo. CMfinder - a covariance model based RNA motif finding algorithm. *Bioinformatics*, 22(4):445–452, 2006.
- [61] M Bauer, GW Klau, and K Reinert. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinformatics*, 8:271, Jul 2007.
- [62] AR Gruber, R Lorenz, SH Bernhart, R Neuböck, and IL Hofacker. The Vienna RNA websuite. *Nucl Acids Res.*, 36(suppl 2):W70–W74, Apr 2008.
- [63] C Smith, S Heyne, AS Richter, S Will, and R Backofen. Freiburg RNA tools: a web server integrating IntaRNA, ExpaRNA and LocARNA. *Nucl Acids Res.*, 38(suppl 2):W373–W377, May 2010.
- [64] K Asai, H Kiryu, M Hamada, Y Tabei, K Sato, H Matsui, Y Sakakibara, G Terai, and T Mituyama. Software.ncrna.org: web servers for analyses of RNA sequences. *Nucl Acids Res.*, 36(suppl2):W75–W78, Apr 2008.
- [65] B Rhead, D Karolchik, RM Kuhn, AS Hinrichs, AS Zweig, PA Fujita, M Diekhands, KE Smith, KR Rosenbloom, BJ Raney, A Pohl, M Pheasant, LR Meyer, K Learned, F Hsu, J Hillman-Jackson, RA Harte, B Giardine, TR Dreszer, H Clawson, GP Barber, D Haussler, and WJ Kent. The UCSC Genome Browser database: update 2010. *Nucl Acids Res.*, 38(suppl 1):D613–D619, Nov 2009.
- [66] P Flicek, MR Amode, D Barrell, K Beal, S Brent, Y Chen, P Clapham, G Coates, S Fairley, S Fitzgerald, L Gordon, M Hendrix, T Hourlier, N Johnson, A Kähäri, D Keefe, S Keenan, R Kinsella, F Kokocinski, E Kulesha, P Larsson, I Longden, W McLaren, B Overduin, B Pritchard, HS Riat, D Rios, GRS Ritchie, M Ruffier, M Schuster, D Sobral, G Spudich, YA Tang, S Trevanion, J Vandrovcova, AJ Vilella, S White, SP Wilder,

- A Zadissa, J Zamora, BL Aken, E Birney, F Cunningham, I Dunham, R Durbin, XM Fernández-Suarez, J Herrero, TJP Hubbard, A Parker, G Proctor, J Vogel, and SMJ Searle. Ensembl 2011. *Nucl Acids Res.*, 39(suppl 1):D800–D806, Nov 2011.
- [67] KL Schneider, KS Pollard, R Baertsch, A Pohl, and TM Lowe. The UCSC Archaeal Genome Browser. *Nucl Acids Res.*, 34(suppl 1):D407–D410, Oct 2005.
- [68] S Griffiths-Jones. Ralee—rna alignment editor in emacs. *Bioinformatics*, 21(2):257–259, Sep 2005.
- [69] J Stombaugh, J Widmann, D McDonald, and R Knight. Boulder ALignment Editor (ale): a web-based RNA alignment tool. *Bioinformatics*, 27(12):1706–1707, Apr 2011.
- [70] F Jossinet and E Westhof. Sequence to structure (S2S): display, manipulate and interconnect RNA data from sequence to structure. *Bioinformatics*, 21(15):3320–3321, May 2005.
- [71] ES Andersen, A Lind-Thomsen, B Knudsen, SE Kristensen, JH Havgaard, E Torarinsson, N Larsen, C Zwieb, P Sestoft, J Kjems, and J Gorodkin. Semiautomated improvement of RNA alignments. *RNA*, 1850-1859(13):1850–1859, Sep 2007.
- [72] E Roberts, J Eargle, D Wright, and Z Luthey-Schulten. Multiseq: unifying sequence and structure data for evolutionary analysis. *BMC Bioinformatics*, 7:382, Aug 2006.
- [73] A M Waterhouse, J B Procter, D M Martin, M Clamp, and G J Barton. Jalview version 2—a multiple sequence alignment editor and analysis workbench. *Bioinformatics*, 25(9):1189–91, May 2009.
- [74] S R Eddy and R Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Res*, 22(11):2079–88, Jun 1994.
- [75] Y Sakakibara, M Brown, R Hughey, I S Mian, K Sjölander, R C Underwood, and D Haussler. Stochastic context-free grammars for tRNA modeling. *Nucleic Acids Res*, 22(23):5112–20, Nov 1994.
- [76] PP Gardner. The use of covariance models to annotate RNAs in whole genomes. *Brief Funct Genomic Proteomic*, 8(6):444–50, Nov 2009.
- [77] EP Nawrocki and SR Eddy. Query-dependent banding (QDB) for faster RNA similarity searches. *PLoS Comput Biol.*, 3(3):e56, Mar 2007.
- [78] EP Nawrocki. Annotating functional RNAs in genomes using Infernal. *Methods in Molecular Biology*. Humana Press, In press 2012.
- [79] MP Hoepfner, L Barquist, and PP Gardner. An introduction to RNA databases. *Methods in Molecular Biology*. Humana Press, In press 2012.
- [80] A Bateman, S Agrawal, E Birney, EA Bruford, JM Bujnicki, et al. RNA-central: A vision for an international database of RNA sequences. *RNA*, 17(11):1941–1946, Sep 2011.

- [81] P Rocca-Serra, S Bellaousov, A Birmingham, C Chen, P Cordero, R Das, L Davis-Neulander, C D Duncan, M Halvorsen, R Knight, N B Leontis, D H Mathews, J Ritz, J Stombaugh, K M Weeks, C L Zirbel, and A Laederach. Sharing and archiving nucleic acid structure mapping data. *RNA*, 17(7):1204–12, Jul 2011.
- [82] JE Stajich, D Block, K Boulez, SE Brenner, et al. The BioPerl toolkit: Perl modules for the life sciences. *Genome Res.*, 12(10):1611–1618, Oct 2002.
- [83] PJ Cock, T Antao, JT Chang, BA Chapman, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, Mar 2009.
- [84] PF Stadler, JJ Chen, J Hackermüller, S Hoffman, F Horn, et al. Evolution of vault RNAs. *Mol Biol Evol.*, 26(9):1975–1991, Sep 2009.
- [85] A Mosig, L Zhu, and PF Stadler. Customized strategies for discovering distant ncRNA homologs. *Brief Funct Genomic Proteomic*, 8(6):451–460, Sep 2009.
- [86] A Mosig, JL Chen, and PF Stadler. Homology search with fragmented nucleic acid sequence patterns. In *Algorithms in Bioinformatics*, volume 4645 of *Lecture Notes in Computer Science*. Springer, 2007.
- [87] J Hertel, D De Jong, M Marz, D Rose, et al. Non-coding RNA annotation of the genome of *Trichoplax adhaerens*. *Nucl Acids Res.*, 37(5):1602–1615, Jan 2009.
- [88] G Storz, JA Opdyke, and A Zhang. Controlling mRNA stability and translation with small, noncoding RNAs. *Curr Opin Microbiol*, 7(2):140–144, Apr 2004.
- [89] EB Gogol, VA Rhodius, K Papenfort, J Vogel, and CA Gross. Small RNAs endow a transcriptional activator with essential repressor functions for single-tier control of a global stress regulon. *PNAS*, 108(31):12875–12880, Jul 2011.
- [90] J Vogel. A rough guide to the non-coding RNA world of *Salmonella*. *Mol Microbiol*, 71(1):1–11, Jan 2009.
- [91] G Storz. *E. coli* small RNAs. http://cbmp.nichd.nih.gov/segr/ecoli_rnas.html, Dec 2011.
- [92] KI Udekwu, F Darfeuille, J Vogel, J Reimegard, et al. Hfq-dependent regulation of OmpA synthesis is mediated by an antisense RNA. *Genes and Dev.*, 19:2355–2366, 2005.